Examiner-Initiated Interview Summary	**Application No.** 10/625,048	**Applicant(s)** BARABASH ET AL.
	Examiner Usmaan Saeed	**Art Unit** 2166

All Participants: **Status of Application:** _____

(1) Usmaan Saeed. (3) _____.

(2) Daniel J Swirsky, Registration No. 45,148 . (4) _____.

Date of Interview: 29 August 2007 **Time:** 12:01 pm

Type of Interview:
☒ Telephonic
□ Video Conference
□ Personal (Copy given to: □ Applicant □ Applicant's representative)

Exhibit Shown or Demonstrated: □ Yes □ No
If Yes, provide a brief description:

Part I.

Rejection(s) discussed:
None

Claims discussed:
12, 20, 42, 44, 46, and 48

Prior art documents discussed:
None

Part II.

SUBSTANCE OF INTERVIEW DESCRIBING THE GENERAL NATURE OF WHAT WAS DISCUSSED:
See Continuation Sheet

Part III.

☒ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview directly resulted in the allowance of the application. The examiner will provide a written summary of the substance of the interview in the Notice of Allowability.

□ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview did not result in resolution of all issues. A brief summary by the examiner appears in Part II above.

(Examiner/SPE Signature) (Applicant/Applicant's Representative Signature – if appropriate)

Continuation of Substance of Interview including description of the general nature of what was discussed: A telephone call was made to applicant's representative about the potential amendments in order to allow the case. The representative agreed with the examiner's proposal and gave authorization for examiner's amendment. A copy of the email from applicant's representative with proposed amendments is also attached.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Applicant:	Katherine BARABASH, et al.
Serial No.:	10/625,048
Filed:	July 23, 2003
Title:	MOSTLY CONCURRENT GARBAGE COLLECTION
Attorney Docket No:	IL920030014US1

Group Art Unit:	2166
Examiner:	U. Saeed

TEXT FOR EXAMINER'S AMENDMENT

TEXT FORPROPOSED EXAMINER'S AMENDMENT – MARKUP FORMAT

1 – 7 (cancelled)

8. (currently amended) A method according to claim 20 and further comprising:

designating any of said objects as "new"; and

deferring the tracing of said "new" objects during any cycle of a plurality of cycles during which any of steps a) – ~~g)~~f) are performed.

9. (original) A method according to claim 8 wherein said designating as "new" step is performed if said object is part of an allocation cache from which objects are currently being allocated.

10. (previously presented) A method according to claim 8 and further comprising:

periodically unmarking any marked card containing only "new" objects; and

periodically removing said "new" objects' "new" designation.

11. (previously presented) A method according to claim 10 wherein said periodically unmarking and periodically removing steps are performed if said object is part of an allocation cache from which objects are not currently being allocated.

12. (currently amended) A method for collecting garbage in a computing environment, the method comprising:

a) tracing a root object to any of its reachable objects in a population of objects;

b) marking any of said objects referred to in step a);

~~d)~~c) tracing any of said marked objects to an unmarked referent object of said marked object;

~~e)~~d) marking said unmarked referent object;

~~f)~~e) tracing said referent object marked in step ~~e)~~d) to any of its reachable objects;

~~g)~~f) marking any of said objects referred to in step ~~f)~~e);

~~h)~~g) tracing any unmarked root object referent to any of its reachable objects;

~~i)~~h) marking any of said objects referred to in step ~~h)~~g);

~~j)~~i) performing any of steps c) – ~~g)~~f); and

~~k)~~j) designating any unmarked object in said population of objects as available for reallocation,

wherein either of steps a) and ~~f)~~e) are performed for a given object only if the card to which the object belongs is not marked, wherein any of steps a) – ~~g)~~f) are performed upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of steps ~~h)~~g) – ~~k)~~j) are performed upon said population of objects while no mutator operates upon said population of objects within said computing environment.

13. (original) A method according to claim 12 and further comprising marking said card if said mutator modifies an object pointer of an object in said card.

14. (currently amended) A method according to claim 12 wherein any of steps a) – ~~g)~~f) are performed concurrently.

15. (currently amended) A method according to claim 12 wherein any of steps ~~h)~~g) – ~~j)~~i) are performed concurrently.

16 - 19 (cancelled)

20. (currently amended) A method for collecting garbage in a computing environment, the method comprising:

a) tracing a root object to any of its reachable objects in a population of objects;

b) marking any of said objects referred to in step a);

~~d)~~c) tracing any of said marked objects to an unmarked referent object of said marked object;

~~e)~~d) marking said unmarked referent object;

~~f)~~e) tracing said referent object marked in step ~~e)~~d) to any of its reachable objects;

~~g)~~f) marking any of said objects referred to in step ~~f)~~e);

~~h)~~g) tracing any unmarked root object referent to any of it's reachable objects;

~~i)~~h) marking any of said objects referred to in step ~~h)~~g);

~~j)~~i) performing any of steps c) – ~~g)~~f);

~~k)~~j) designating any unmarked object in said population of objects as available for reallocation; and

~~l)~~k) at any time while performing concurrently any of steps a) – ~~g)~~f), periodically determining whether a marked card contains at least one of said marked objects, and unmarking any marked card about which it is determined that it does not contain at least one of said marked objects,

wherein any of steps a) – ~~g)~~f) are performed upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of steps ~~h)~~g) – ~~k)~~j) are performed upon said population of objects while no mutator operates upon said population of objects within said computing environment.

21. (original) A method according to claim 20 and further comprising marking said card if said mutator modifies an object pointer of an object in said card.

22. (currently amended) A method according to claim 20 wherein any of steps a) – ~~g)~~f) are performed concurrently.

23. (currently amended) A method according to claim 20 wherein any of steps ~~h)~~g) – ~~j)~~i) are performed concurrently.

24 – 37 (cancelled)

38. (currently amended) A system according to claim 44 and further comprising:

means for designating any of said objects as "new"; and means for deferring the tracing of said "new" objects during any cycle of a plurality of cycles during which any of means a) – ~~g)~~f) operate.

39. (original) A system according to claim 38 wherein said means for designating as "new" is operative if said object is part of an allocation cache from which objects are currently being allocated.

40. (original) A system according to claim 38 and further comprising:

means for periodically unmarking any marked card containing only "new" objects; and

means for removing said "new" objects' "new" designation.

41. (original) A system according to claim 40 wherein said means for periodically unmarking and said means for removing are operative if said object is part of an allocation cache from which objects are not currently being allocated.

42. (currently amended) A system for collecting garbage in a computing environment, the system comprising:

a) means for tracing a root object to any of its reachable objects in a population of objects;

b) means for marking any of said objects referred to in a);

~~d)~~c) means for tracing any of said marked objects to an unmarked referent object of said marked object;

~~e)~~d) means for marking said unmarked referent object;

~~f)~~e) means for tracing said marked referent object in ~~e)~~d) to any of its reachable objects;

~~g)~~f) means for marking any of said objects referred to in ~~f)~~e);

~~h)~~g) means for tracing any unmarked root object referent to any of its reachable objects;

~~i)~~h) means for marking any of said objects referred to in ~~h)~~g); and

~~j)~~i) means for designating any unmarked object in said population of objects as available for reallocation, wherein either of tracing means a) and ~~f)~~e) trace a given object only if the card to which the object belongs is not marked, wherein any of means a) – ~~g)~~f) operate upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of means ~~h)~~g) – ~~j)~~i) operate upon said population of objects while no mutator operates upon said population of objects within said computing environment.

43. (cancelled)

44. (currently amended) A system for collecting garbage in a computing environment, the system comprising:

a) means for tracing a root object to any of its reachable objects in a population of objects;

b) means for marking any of said objects referred to in a);

~~d)~~c) means for tracing any of said marked objects to an unmarked referent object of said marked object;

~~e)~~d) means for marking said unmarked referent object;

~~f)~~e) means for tracing said marked referent object in ~~e)~~d) to any of its reachable objects;

~~g)~~f) means for marking any of said objects referred to in ~~f)~~e);

~~h)~~g) means for tracing any unmarked root object referent to any of its reachable objects;

~~i)~~h) means for marking any of said objects referred to in ~~h)~~g);

~~j)~~i) means for designating any unmarked object in said population of objects as available for reallocation; and

~~k)~~j) means for periodically determining at any time concurrently with the operation of any of means a) – ~~g)~~f) whether a marked card contains at least one of said marked objects, and unmarking any marked card about which it is determined that it does not contain at least one of said marked objects,

wherein any of means a) – ~~g)~~f) operate upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of means ~~h)~~g) – ~~j)~~i) operate upon said population of objects while no mutator operates upon said population of objects within said computing environment.

45. (cancelled)

46. (currently amended) A computer program embodied on a computer-readable medium, the computer program comprising:

a) a first code segment operative to trace a root object to any of its reachable objects in a population of objects;

b) a second code segment operative to mark any of said objects referred to in a);

~~d)~~c) a ~~fourth~~third code segment operative to trace any of said marked objects to an unmarked referent object of said marked object;

~~e)~~d) a ~~fifth~~fourth code segment operative to mark said unmarked referent object;

~~f)~~e) a ~~sixth~~fifth code segment operative to trace said marked referent object in ~~e)~~d) to any of its reachable objects;

~~g)~~f) a ~~seventh~~sixth code segment operative to mark any of said objects referred to in ~~f)~~e);

~~h)~~g) a ~~eighth~~seventh code segment operative to trace any unmarked root object referent to any of its reachable objects;

~~i)~~h) a ~~ninth~~eighth code segment operative to mark any of said objects referred to in ~~h)~~g); and

~~j)~~i) a ~~tenth~~ninth code segment operative to designate any unmarked object in said population of objects as available for reallocation,

wherein either of said code segments a) and ~~f)~~e) are operative to trace a given object only if the card to which the object belongs is not marked, wherein any of said code segments a) -- ~~g)~~f) operate upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of said code segments ~~h)~~g) – ~~j)~~i) operate upon said population of objects while no mutator operates upon said population of objects within said computing environment.

47. (cancelled)

48. (currently amended) A method for collecting garbage in a computing environment, the method comprising:

a) a first code segment operative to trace a root object to any of its reachable objects in a population of objects;

b) a second code segment operative to mark any of said objects referred to in a);

~~d)~~c) a ~~fourth~~third code segment operative to trace any of said marked objects to an unmarked referent object of said marked object;

~~e)~~d) a ~~fifth~~fourth code segment operative to mark said unmarked referent object;

~~f)~~e) a ~~sixth~~fifth code segment operative to trace said marked referent object in ~~e)~~d) to any of its reachable objects;

~~g)~~f) a ~~seventh~~sixth code segment operative to mark any of said objects referred to in ~~f)~~e);

~~h)~~g) a ~~eighth~~seventh code segment operative to trace any unmarked root object referent to any of its reachable objects;

~~i)~~h) a ~~ninth~~eighth code segment operative to mark any of said objects referred to in ~~h)~~g);

~~j)~~i) a ~~tenth~~ninth code segment operative to designate any unmarked object in said population of objects as available for reallocation; and

~~k)~~j) an ~~eleventh~~tenth code segment operative at any time concurrent to the operation of any of said code segments a) – ~~g)~~f), to periodically determine whether a marked card contains at least one of said marked objects, and unmark any marked card about which it is determined that it does not contain at least one of said marked objects,

wherein any of said code segments a) – ~~g)~~f) are operative upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of said code segments ~~h)~~g) – ~~j)~~i) are operative upon said population of objects while no mutator operates upon said population of objects within said computing environment.

49. (cancelled)

TEXT FOR PROPOSED EXAMINER'S AMENDMENT – WITHOUT MARKINGS

1 – 7 (cancelled)

8. A method according to claim 20 and further comprising:

designating any of said objects as "new"; and

deferring the tracing of said "new" objects during any cycle of a plurality of cycles during which any of steps a) – f) are performed.

9. A method according to claim 8 wherein said designating as "new" step is performed if said object is part of an allocation cache from which objects are currently being allocated.

10. A method according to claim 8 and further comprising:

periodically unmarking any marked card containing only "new" objects; and

periodically removing said "new" objects' "new" designation.

11. A method according to claim 10 wherein said periodically unmarking and periodically removing steps are performed if said object is part of an allocation cache from which objects are not currently being allocated.

12. A method for collecting garbage in a computing environment, the method comprising:

a) tracing a root object to any of its reachable objects in a population of objects;

b) marking any of said objects referred to in step a);

c) tracing any of said marked objects to an unmarked referent object of said marked object;

d) marking said unmarked referent object;

e) tracing said referent object marked in step d) to any of its reachable objects;

f) marking any of said objects referred to in step e);

g) tracing any unmarked root object referent to any of its reachable objects;

h) marking any of said objects referred to in step g);

i) performing any of steps c) – f); and

j) designating any unmarked object in said population of objects as available for reallocation,

wherein either of steps a) and e) are performed for a given object only if the card to which the object belongs is not marked, wherein any of steps a) – f) are performed upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of steps g) – j) are performed upon said population of objects while no mutator operates upon said population of objects within said computing environment.

13. A method according to claim 12 and further comprising marking said card if said mutator modifies an object pointer of an object in said card.

14. A method according to claim 12 wherein any of steps a) – f) are performed concurrently.

15. A method according to claim 12 wherein any of steps g) – i) are performed concurrently.

16 - 19 (cancelled)

20. A method for collecting garbage in a computing environment, the method comprising:

a) tracing a root object to any of its reachable objects in a population of objects;

b) marking any of said objects referred to in step a);

c) tracing any of said marked objects to an unmarked referent object of said marked object;

d) marking said unmarked referent object;

e) tracing said referent object marked in step d) to any of its reachable objects;

f) marking any of said objects referred to in step e);

g) tracing any unmarked root object referent to any of its reachable objects;

h) marking any of said objects referred to in step g);

i) performing any of steps c) – f);

j) designating any unmarked object in said population of objects as available for reallocation; and

k) at any time while performing concurrently any of steps a) – f), periodically determining whether a marked card contains at least one of said marked objects, and unmarking any marked card about which it is determined that it does not contain at least one of said marked objects,

wherein any of steps a) – f) are performed upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of steps g) – j) are performed upon said population of objects while no mutator operates upon said population of objects within said computing environment.

21. A method according to claim 20 and further comprising marking said card if said mutator modifies an object pointer of an object in said card.

22. A method according to claim 20 wherein any of steps a) – f) are performed concurrently.

23. A method according to claim 20 wherein any of steps g) – i) are performed concurrently.

24 – 37 (cancelled)

38. A system according to claim. 44 and further comprising:

means for designating any of said objects as “new”; and

means for deferring the tracing of said “new” objects during any cycle of a plurality of cycles during which any of means a) – f) operate.

39. A system according to claim 38 wherein said means for designating as “new” is operative if said object is part of an allocation cache from which objects are currently being allocated.

40. A system according to claim 38 and further comprising:

means for periodically unmarking any marked card containing only “new” objects; and

means for removing said “new” objects’ “new” designation.

41. A system according to claim 40 wherein said means for periodically unmarking and said means for removing are operative if said object is part of an allocation cache from which objects are not currently being allocated.

42. A system for collecting garbage in a computing environment, the system comprising:

a) means for tracing a root object to any of its reachable objects in a population of objects;

b) means for marking any of said objects referred to in a);

c) means for tracing any of said marked objects to an unmarked referent object of said marked object;

d) means for marking said unmarked referent object;

e) means for tracing said marked referent object in d) to any of its reachable objects;

f) means for marking any of said objects referred to in e);

g) means for tracing any unmarked root object referent to any of its reachable objects;

h) means for marking any of said objects referred to in g); and

i) means for designating any unmarked object in said population of objects as available for reallocation,

wherein either of tracing means a) and e) trace a given object only if the card to which the object belongs is not marked, wherein any of means a) – f) operate upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of means g) – i) operate upon said population of objects while no mutator operates upon said population of objects within said computing environment.

43. (cancelled)

44. A system for collecting garbage in a computing environment, the system comprising:

a) means for tracing a root object to any of its reachable objects in a population of objects;

b) means for marking any of said objects referred to in a);

c) means for tracing any of said marked objects to an unmarked referent object of said marked object;

d) means for marking said unmarked referent object;

e) means for tracing said marked referent object in d) to any of its reachable objects;

f) means for marking any of said objects referred to in e);

g) means for tracing any unmarked root object referent to any of its reachable objects;

h) means for marking any of said objects referred to in g);

i) means for designating any unmarked object in said population of objects as available for reallocation; and

j) means for periodically determining at any time concurrently with the operation of any of means a) – f) whether a marked card contains at least one of said marked objects, and unmarking any marked card about which it is determined that it does not contain at least one of said marked objects, wherein any of means a) – f) operate upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of means g) – i) operate upon said population of objects while no mutator operates upon said population of objects within said computing environment.

45. (cancelled)

46. A computer program embodied on a computer-readable medium, the computer program comprising:

a) a first code segment operative to trace a root object to any of its reachable objects in a population of objects;

b) a second code segment operative to mark any of said objects referred to in a);

c) a third code segment operative to trace any of said marked objects to an unmarked referent object of said marked object;

d) a fourth code segment operative to mark said unmarked referent object;

e) a fifth code segment operative to trace said marked referent object in d) to any of its reachable objects;

f) a sixth code segment operative to mark any of said objects referred to in e);

g) a seventh code segment operative to trace any unmarked root object referent to any of its reachable objects;

h) a eighth code segment operative to mark any of said objects referred to in g); and

i) a ninth code segment operative to designate any unmarked object in said population of objects as available for reallocation,

wherein either of said code segments a) and e) are operative to trace a given object only if the card to which the object belongs is not marked, wherein any of said code segments a) – f) operate upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of said code segments g) – i) operate upon said population of objects while no mutator operates upon said population of objects within said computing environment.

47. (cancelled)

48. A method for collecting garbage in a computing environment, the method comprising:

a) a first code segment operative to trace a root object to any of its reachable objects in a population of objects;

b) a second code segment operative to mark any of said objects referred to in a);

c) a third code segment operative to trace any of said marked objects to an unmarked referent object of said marked object;

d) a fourth code segment operative to mark said unmarked referent object;

e) a fifth code segment operative to trace said marked referent object in d) to any of its reachable objects;

f) a sixth code segment operative to mark any of said objects referred to in e);

g) a seventh code segment operative to trace any unmarked root object referent to any of its reachable objects;

h) a eighth code segment operative to mark any of said objects referred to in g);

i) a ninth code segment operative to designate any unmarked object in said population of objects as available for reallocation; and

j) an tenth code segment operative at any time concurrent to the operation of any of said code segments a) – f), to periodically determine whether a marked card contains at least one of said marked objects, and unmark any marked card about which it is determined that it does not contain at least one of said marked objects,

wherein any of said code segments a) – f) are operative upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of said code segments g) – i) are operative upon said population of objects while no mutator operates upon said population of objects within said computing environment.

49. (cancelled)

TEXT FOR PROPOSED EXAMINER'S AMENDMENT – RENUMBERED

1. A method for collecting garbage in a computing environment, the method comprising:

a) tracing a root object to any of its reachable objects in a population of objects;

b) marking any of said objects referred to in step a);

c) tracing any of said marked objects to an unmarked referent object of said marked object;

d) marking said unmarked referent object;

e) tracing said referent object marked in step d) to any of its reachable objects;

f) marking any of said objects referred to in step e);

g) tracing any unmarked root object referent to any of its reachable objects;

h) marking any of said objects referred to in step g);

i) performing any of steps c) – f); and

j) designating any unmarked object in said population of objects as available for reallocation,

wherein either of steps a) and e) are performed for a given object only if the card to which the object belongs is not marked, wherein any of steps a) – f) are performed upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of steps g) – j) are performed upon said population of objects while no mutator operates upon said population of objects within said computing environment.

2. A method according to claim 1 and further comprising marking said card if said mutator modifies an object pointer of an object in said card.

3. A method according to claim 1 wherein any of steps a) – f) are performed concurrently.

4. A method according to claim 1 wherein any of steps g) – i) are performed concurrently.

5. A method for collecting garbage in a computing environment, the method comprising:

a) tracing a root object to any of its reachable objects in a population of objects;

b) marking any of said objects referred to in step a);

c) tracing any of said marked objects to an unmarked referent object of said marked object;

d) marking said unmarked referent object;

e) tracing said referent object marked in step d) to any of its reachable objects;

f) marking any of said objects referred to in step e);

g) tracing any unmarked root object referent to any of its reachable objects;

h) marking any of said objects referred to in step g);

i) performing any of steps c) – f);

j) designating any unmarked object in said population of objects as available for reallocation; and

k) at any time while performing concurrently any of steps a) – f), periodically determining whether a marked card contains at least one of said marked objects, and unmarking any marked card about which it is determined that it does not contain at least one of said marked objects,

wherein any of steps a) – f) are performed upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of steps g) -- j) are performed upon said population of objects while no mutator operates upon said population of objects within said computing environment.

6. A method according to claim 5 and further comprising:

designating any of said objects as "new"; and deferring the tracing of said "new" objects during any cycle of a plurality of cycles during which any of steps a) – f) are performed.

7. A method according to claim 6 wherein said designating as "new" step is performed if said object is part of an allocation cache from which objects are currently being allocated.

8. A method according to claim 6 and further comprising:

periodically unmarking any marked card containing only "new" objects; and periodically removing said "new" objects' "new" designation.

9. A method according to claim 8 wherein said periodically unmarking and periodically removing steps are performed if said object is part of an allocation cache from which objects are not currently being allocated.

10. A method according to claim 5 and further comprising marking said card if said mutator modifies an object pointer of an object in said card.

11. A method according to claim 5 wherein any of steps a) – f) are performed concurrently.

12. A method according to claim 5 wherein any of steps g) – i) are performed concurrently.

13. A system for collecting garbage in a computing environment, the system comprising:

a) means for tracing a root object to any of its reachable objects in a population of objects;

b) means for marking any of said objects referred to in a);

c) means for tracing any of said marked objects to an unmarked referent object of said marked object;

d) means for marking said unmarked referent object;

e) means for tracing said marked referent object in d) to any of its reachable objects;

f) means for marking any of said objects referred to in e);

g) means for tracing any unmarked root object referent to any of its reachable objects;

h) means for marking any of said objects referred to in g); and

i) means for designating any unmarked object in said population of objects as available for reallocation,

wherein either of tracing means a) and e) trace a given object only if the card to which the object belongs is not marked, wherein any of means a) – f) operate upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of means g) – i) operate upon said population of objects while no mutator operates upon said population of objects within said computing environment.

14. A system for collecting garbage in a computing environment, the system comprising:

a) means for tracing a root object to any of its reachable objects in a population of objects;

b) means for marking any of said objects referred to in a);

c) means for tracing any of said marked objects to an unmarked referent object of said marked object;

d) means for marking said unmarked referent object;

e) means for tracing said marked referent object in d) to any of its reachable objects;

f) means for marking any of said objects referred to in e);

g) means for tracing any unmarked root object referent to any of its reachable objects;

h) means for marking any of said objects referred to in g);

i) means for designating any unmarked object in said population of objects as available for reallocation; and

j) means for periodically determining at any time concurrently with the operation of any of means a) – f) whether a marked card contains at least one of said marked objects, and unmarking any marked card about which it is determined that it does not contain at least one of said marked objects,

wherein any of means a) – f) operate upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of means g) – i) operate upon said population of objects while no mutator operates upon said population of objects within said computing environment.

15. A system according to claim 14 and further comprising:

means for designating any of said objects as "new"; and

means for deferring the tracing of said "new" objects during any cycle of a plurality of cycles during which any of means a) – f) operate.

16. A system according to claim 15 wherein said means for designating as "new" is operative if said object is part of an allocation cache from which objects are currently being allocated.

17. A system according to claim 15 and further comprising:

means for periodically unmarking any marked card containing only "new" objects; and

means for removing said "new" objects' "new" designation.

18. A system according to claim 17 wherein said means for periodically unmarking and said means for removing are operative if said object is part of an allocation cache from which objects are not currently being allocated.

19. A computer program embodied on a computer-readable medium, the computer program comprising:

a) a first code segment operative to trace a root object to any of its reachable objects in a population of objects;

b) a second code segment operative to mark any of said objects referred to in a);

c) a third code segment operative to trace any of said marked objects to an unmarked referent object of said marked object;

d) a fourth code segment operative to mark said unmarked referent object;

e) a fifth code segment operative to trace said marked referent object in d) to any of its reachable objects;

f) a sixth code segment operative to mark any of said objects referred to in e);

g) a seventh code segment operative to trace any unmarked root object referent to any of its reachable objects;

h) a eighth code segment operative to mark any of said objects referred to in g); and

i) a ninth code segment operative to designate any unmarked object in said population of objects as available for reallocation,

wherein either of said code segments a) and e) are operative to trace a given object only if the card to which the object belongs is not marked, wherein any of said code segments a) – f) operate upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of said code segments g) – i) operate upon said population of objects while no mutator operates upon said population of objects within said computing environment.

20. A method for collecting garbage in a computing environment, the method comprising:

a) a first code segment operative to trace a root object to any of its reachable objects in a population of objects;

b) a second code segment operative to mark any of said objects referred to in a);

c) a third code segment operative to trace any of said marked objects to an unmarked referent object of said marked object;

d) a fourth code segment operative to mark said unmarked referent object;

e) a fifth code segment operative to trace said marked referent object in d) to any of its reachable objects;

f) a sixth code segment operative to mark any of said objects referred to in e);

g) a seventh code segment operative to trace any unmarked root object referent to any of its reachable objects;

h) a eighth code segment operative to mark any of said objects referred to in g);

i) a ninth code segment operative to designate any unmarked object in said population of objects as available for reallocation; and

j) an tenth code segment operative at any time concurrent to the operation of any of said code segments a) – f), to periodically determine whether a marked card contains at least one of said marked objects, and unmark any marked card about which it is determined that it does not contain at least one of said marked objects,

wherein any of said code segments a) – f) are operative upon said population of objects concurrently with the operation of a mutator upon said population of objects within said computing environment, and wherein any of said code segments g) – i) are operative upon said population of objects while no mutator operates upon said population of objects within said computing environment.